Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	1005	transaction? with priorit$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:07
L2	420448	journal or journaling or log or logging	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:07
L3	7559	(journal or journaling or log or logging) with (database)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:08
L4	58	1 and 3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:08
L5	12040459	@ad<"20011213"	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:08
L6	47	4 and 5	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:11
L7	1216	deallocat$3 near3 memory	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:11
L8	1	(adjust$3 adj2 (transaction adj priority)) or (modif$7 adj2 (transaction adj priority))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:16
L9	1	(adjust$3 with (transaction adj priority))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:16
L10	455	priorit$3 near3 transaction?	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:17
L11	3	7 and 10	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:17
L12	2	5 and 11	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:18
L13	35	journal adj space	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:18

L14	2	(reduce or reduced or reducing) with L13	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:21
L15	1	1 and 13	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:19
L16	16	(reduce or reduced or reducing) with (log adj space or log adj size)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:20
L17	0	1 and 16	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:21
L18	4	(reduce or reduced or reducing) with journal adj size	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:22
L19	1	(US-6189016-$).did.	USPAT	OR	OFF	2004/11/13 14:22
L20	1	10 and 18	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:24
L21	20	(transaction adj history or transaction adj order) with (optimiz$3 or priorit$3)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:25
L22	17	5 and 21	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:26
L23	2	2 and 22	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:26
L24	538	711/158.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:27
L25	646	711/111.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:27
L26	1646	711/154.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:28
L27	1376	711/170.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:28

L28	3881	24 or 25 or 26 or 27	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:28
L29	1	4 and 28	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:28
L30	1828	707/2.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:28
L31	752	707/202.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:28
L32	2551	30 or 31	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:28
L33	5	4 and 32	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:28
L34	5	5 and 33	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:29
L35	1327	705/35.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:29
L36	1130	705/39.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:30
L37	306	705/42.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:30
L38	2511	35 or 36 or 37	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:30
L39	2	4 and 38	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:30
L40	0	(journal adj space) with allocat$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:31
L41	297	(journal) with allocat$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:32

L42	3	1 and 41	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:31
L43	3	(journal) with deallocat$3	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:32
L44	0	1 and 43	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2004/11/13 14:32

Your search matched **23** of **1091947** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance** in **Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or entering a new one in the text box.

journal <and> (allocation <or> deallocation) <and> prio Search

Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

---

1 **A multiservice traffic allocation model for LEO satellite communication networks**
McMahon, G.; Septiawan, R.; Sugden, S.;
Selected Areas in Communications, IEEE Journal on , Volume: 22 , Issue: 3 , April 2004
Pages:501 - 507

[Abstract] [PDF Full-Text (312 KB)] IEEE JNL

---

2 **A guided-wave optical arbiter circuit**
Shimizu, N.; Ikeda, M.; Imase, M.; Okada, K.; Kimura, T.;
Lightwave Technology, Journal of , Volume: 1 , Issue: 2 , Jun 1983
Pages:424 - 429

[Abstract] [PDF Full-Text (2336 KB)] IEEE JNL

---

3 **Dynamic reservation TDMA protocol for wireless ATM networks**
Frigon, J.-F.; Leung, V.C.M.; Chan Bun Chan, H.;
Selected Areas in Communications, IEEE Journal on , Volume: 19 , Issue: 2 , Feb 2001
Pages:370 - 383

[Abstract] [PDF Full-Text (232 KB)] IEEE JNL

---

4 **Performance of a priority-based dynamic capacity allocation scheme for wireless ATM systems**
Ganesh Babu, T.V.J.; Le-Ngoc, T.; Hayes, J.F.;
Selected Areas in Communications, IEEE Journal on , Volume: 19 , Issue: 2 , Feb 2001
Pages:355 - 369

[Abstract] [PDF Full-Text (268 KB)] IEEE JNL

---

5 **A nonpreemptive priority-based access control scheme for broadband ad hoc wireless ATM local area networks**
Dr-Jiunn Deng; Ruay-Shiung Chang;
Selected Areas in Communications, IEEE Journal on , Volume: 18 , Issue: 9 , Sept. 2000
Pages:1731 - 1739

[Abstract] [PDF Full-Text (268 KB)] IEEE JNL

---

6 **Spectrum resource allocation for wireless packet access with application to advanced cellular Internet service**
Chuang, J.C.-I.; Sollenberger, N.R.;
Selected Areas in Communications, IEEE Journal on , Volume: 16 , Issue: 6 , Aug. 1998

Pages:820 - 829

[Abstract] [PDF Full-Text (176 KB)] IEEE JNL

---

7 **The no slot wasting bandwidth balancing mechanism for dual bus architectures**
Karvelas, D.; Papamichail, M.;
Selected Areas in Communications, IEEE Journal on , Volume: 11 , Issue: 8 , Oct. 1993
Pages:1214 - 1228

[Abstract] [PDF Full-Text (1552 KB)] IEEE JNL

---

8 **A dynamic bandwidth-allocation-based priority mechanism for the p_i-persistent protocol for MAN's**
Miller, G.J.; Paterakis, M.;
Selected Areas in Communications, IEEE Journal on , Volume: 11 , Issue: 8 , Oct. 1993
Pages:1229 - 1239

[Abstract] [PDF Full-Text (1000 KB)] IEEE JNL

---

9 **Dual bus MAN's with multiple-priority traffic**
Liebeherr, J.; Akyildiz, I.F.; Tantawi, A.N.;
Selected Areas in Communications, IEEE Journal on , Volume: 11 , Issue: 8 , Oct. 1993
Pages:1202 - 1213

[Abstract] [PDF Full-Text (856 KB)] IEEE JNL

---

10 **Priority queueing strategies and buffer allocation protocols for traffic control at an ATM integrated broadband switching system**
Lin, A.Y.-M.; Silvester, J.A.;
Selected Areas in Communications, IEEE Journal on , Volume: 9 , Issue: 9 , Dec. 1991
Pages:1524 - 1536

[Abstract] [PDF Full-Text (948 KB)] IEEE JNL

---

11 **An ATM switch architecture for provision of integrated broadband services**
Pattavina, A.;
Selected Areas in Communications, IEEE Journal on , Volume: 9 , Issue: 9 , Dec. 1991
Pages:1537 - 1548

[Abstract] [PDF Full-Text (1076 KB)] IEEE JNL

---

12 **A Novel Decentralized Ethernet-Based PON Access Architecture for Provisioning Differentiated QoS**
Sherif, S.R.; Hadjiantonis, A.; Ellinas, G.; Assi, C.; Ali, M.A.;
Lightwave Technology, Journal of , Volume: 22 , Issue: 11 , Nov. 2004
Pages:2483 - 2497

[Abstract] [PDF Full-Text (1112 KB)] IEEE JNL

---

13 **Switching for IP-based multimedia satellite communications**
Tho Le-Ngoc;
Selected Areas in Communications, IEEE Journal on , Volume: 22 , Issue: 3 , April 2004
Pages:462 - 471

[Abstract] [PDF Full-Text (440 KB)] IEEE JNL

---

14 **Fast Switching and Fair Control of Congested Flow in Broadband Networks**
Katevenis, M.;
Selected Areas in Communications, IEEE Journal on , Volume: 5 , Issue: 8 , Oct 1987
Pages:1315 - 1326

[Abstract] [PDF Full-Text (1336 KB)] IEEE JNL

---

15 **Integration of Voice and Data in the Wideband Packet Satellite Network**

Falk, G.; Groff, J.; Milliken, W.; Nodine, M.; Blumenthal, S.; Edmond, W.;
Selected Areas in Communications, IEEE Journal on , Volume: 1 , Issue: 6 , Dec 1983
Pages:1076 - 1083

[Abstract] [PDF Full-Text (1032 KB)] **IEEE JNL**

Welcome to IEEE Xplore

- Home
- What Can I Access?
- Log-out

Tables of Contents

- Journals & Magazines
- Conference Proceedings
- Standards

Search

- By Author
- Basic
- Advanced
- CrossRef

Member Services

- Join IEEE
- Establish IEEE Web Account
- Access the IEEE Member Digital Library

IEEE Enterprise

- Access the IEEE Enterprise File Cabinet

Print Format

Your search matched **23** of **1091947** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance** in **Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or entering a new one in the text box.

journal <and> (allocation <or> deallocation) <and> prio[Search

Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

---

16 **A framework for bandwidth degradation and call admission control schemes for multiclass traffic in next-generation wireless networks**
Das, S.K.; Sen, S.K.; Basu, K.; Haitao Lin;
Selected Areas in Communications, IEEE Journal on , Volume: 21 , Issue: 10 , Dec. 2003
Pages:1790 - 1802

[Abstract] [PDF Full-Text (732 KB)] **IEEE JNL**

---

17 **Output and delay process analysis for slotted CDMA wireless communication networks with integrated voice/data transmission**
Wuyi Yue; Matsumoto, Y.;
Selected Areas in Communications, IEEE Journal on , Volume: 18 , Issue: 7 , July 2000
Pages:1245 - 1253

[Abstract] [PDF Full-Text (224 KB)] **IEEE JNL**

---

18 **Efficient sequencing techniques for variable-length messages in WDM networks**
Hamidzadeh, B.; Maode, M.; Hamdi, M.;
Lightwave Technology, Journal of , Volume: 17 , Issue: 8 , Aug. 1999
Pages:1309 - 1319

[Abstract] [PDF Full-Text (160 KB)] **IEEE JNL**

---

19 **Virtual partitioning for robust resource sharing: computational techniques for heterogeneous traffic**
Borst, S.C.; Mitra, D.;
Selected Areas in Communications, IEEE Journal on , Volume: 16 , Issue: 5 , June 1998
Pages:668 - 678

[Abstract] [PDF Full-Text (632 KB)] **IEEE JNL**

---

20 **Adaptive resource allocation for prioritized call admission over an ATM-based wireless PCN**
Yu, O.T.W.; Leung, V.C.M.;
Selected Areas in Communications, IEEE Journal on , Volume: 15 , Issue: 7 , Sept. 1997
Pages:1208 - 1225

[Abstract] [PDF Full-Text (476 KB)] **IEEE JNL**

---

21 **Performance evaluation of a Batcher-banyan interconnection network with output pooling**

Pattavina, A.;
Selected Areas in Communications, IEEE Journal on , Volume: 9 , Issue: 1 , Jan. 1991
Pages:95 - 103

[Abstract] [PDF Full-Text (776 KB)] IEEE JNL

---

22 **A call admission control scheme for ATM networks using a simple quality estimate**
Murase, T.; Suzuki, H.; Sato, S.; Takeuchi, T.;
Selected Areas in Communications, IEEE Journal on , Volume: 9 , Issue: 9 , Dec. 1991
Pages:1461 - 1470

[Abstract] [PDF Full-Text (804 KB)] IEEE JNL

---

23 **Flow control schemes and delay/loss tradeoff in ATM networks**
Ohnishi, H.; Okada, T.; Noguchi, K.;
Selected Areas in Communications, IEEE Journal on , Volume: 6 , Issue: 9 , Dec. 1988
Pages:1609 - 1616

[Abstract] [PDF Full-Text (660 KB)] IEEE JNL

Prev 1 2

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search | Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting | No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ | Terms | Back to Top

Terms used
journal and **allocate** or **allocation** or **deallocate** or **deallocation** and **transaction priority**

Found **21,006** of **145,831**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 Provably efficient scheduling for languages with fine-grained parallelism

Guy E. Blelloch, Phillip B. Gibbons, Yossi Matias

March 1999 **Journal of the ACM (JACM)**, Volume 46 Issue 2

Full text available: pdf(321.43 KB) Additional Information: full citation, abstract, references, citings, index terms

Many high-level parallel programming languages allow for fine-grained parallelism. As in the popular work-time framework for parallel algorithm design, programs written in such languages can express the full parallelism in the program without specifying the mapping of program tasks to processors. A common concern in executing such programs is to schedule tasks to processors dynamically so as to minimize not only the execution time, but also the amount of space (memory) needed. Without caref ...

2 A new "quad-tree-based" sub-system allocation technique for mesh-connected parallel machines

Jeeraporn Srisawat, Nikitas A. Alexandridis

May 1999 **Proceedings of the 13th international conference on Supercomputing**

Full text available: pdf(1.22 MB) Additional Information: full citation, references, index terms

Keywords: mesh-connected parallel machines, parallel systems, quad-trees, resource allocation/deallocation, system partitioning

3 Processor allocation for a class of hypercube-like supercomputers

N. G. Haravu, S. G. Ziavras

December 1992 **Proceedings of the 1992 ACM/IEEE conference on Supercomputing**

Full text available: pdf(852.24 KB) Additional Information: full citation, references, citings, index terms

4 Integrating object-oriented programming and protected objects in Ada 95

A. J. Wellings, B. Johnson, B. Sanden, J. Kienzle, T. Wolf, S. Michell

May 2000 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 22 Issue 3

Full text available: pdf(245.47 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Integrating concurrent and object-oriented programming has been an active research topic since the late 1980's. There is a now a plethora of methods for achieving this integration. The majority of approaches have taken a sequential object-oriented language and made it

concurrent. A few approaches have taken a concurrent language and made it object-oriented. The most important of this latter class is the Ada 95 language, which is an extension to the object-based concurrent programming langua ...

Keywords: Ada 95, concurrency, concurrent object-oriented programming, inheritance anomaly

5 Space-efficient scheduling of nested parallelism

Girija J. Narlikar, Guy E. Blelloch

January 1999 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 21 Issue 1

Full text available: pdf(481.02 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Many of today's high-level parallel languages support dynamic, fine-grained parallelism. These languages allow the user to expose all the parallelism in the program, which is typically of a much higher degree than the number of processors. Hence an efficient scheduling algorithm is required to assign computations to processors at runtime. Besides having low overheads and good load balancing, it is important for the scheduling algorithm to minimize the space usage of the parallel program. T ...

Keywords: dynamic scheduling, multithreading, nested parallelism, parallel language implementation, space efficiency

6 Integrating object-oriented programming and protected objects in Ada 95

A. J. Wellings, B. Johnson, B. Sanden, J. Kienzle, T. Wolf, S. Michell

June 2002 **ACM SIGAda Ada Letters**, Volume XXII Issue 2

Full text available: pdf(2.09 MB) Additional Information: full citation, abstract, references, citings, index terms

Integrating concurrent and object-oriented programming has been an active research topic since the late 1980's. There is now a plethora of methods for achieving this integration. The majority of approaches have taken a sequential object-oriented language and made it concurrent. A few approaches have taken a concurrent language and made it object-oriented. The most important of this latter class is the Ada 95 language, which is an extension to the object-based concurrent programming language Ada ...

Keywords: Ada 95, concurrency, concurrent object-oriented programming, inheritance anomaly

7 Space/time-efficient scheduling and execution of parallel irregular computations

Tao Yang, Cong Fu

November 1998 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 20 Issue 6

Full text available: pdf(374.95 KB) Additional Information: full citation, abstract, references, citings, index terms

In this article we investigate the trade-off between time and space efficiency in scheduling and executing parallel irregular computations on distributed-memory machines. We employ acyclic task dependence graphs to model irregular parallelism with mixed granularity, and we use direct remote memory access to support fast communication. We propose new scheduling techniques and a run-time active memory management scheme to improve memory utilization while retaining good time efficiency, and we ...

Keywords: DAG scheduling, direct remote memory access, irregular parallelism, run-time support

8 Efficient implementation of the first-fit strategy for dynamic storage allocation

R. P. Brent
July 1989 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 11 Issue 3

Full text available: pdf(1.05 MB) Additional Information: full citation, abstract, references, citings, index terms, review

We describe an algorithm that efficiently implements the first-fit strategy for dynamic storage allocation. The algorithm imposes a storage overhead of only one word per allocated block (plus a few percent of the total space used for dynamic storage), and the time required to allocate or free a block is O(log W), where W is the maximum number of words allocated dynamically. The algorithm is faster than many commonly used algorithms, especia ...

9 On local register allocation

Martin Farach, Vincenzo Liberatore
January 1998 **Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms**

Full text available: pdf(1.03 MB) Additional Information: full citation, references, citings, index terms

10 Space-efficient scheduling of parallelism with synchronization variables

Guy E. Blelloch, Phillip B. Gibbons, Girija J. Narlikar, Yossi Matias
June 1997 **Proceedings of the ninth annual ACM symposium on Parallel algorithms and architectures**

Full text available: pdf(1.67 MB) Additional Information: full citation, references, citings, index terms

11 Compilation and run-time systems: A faster optimal register allocator

Changqing Fu, Kent Wilken
November 2002 **Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture**

Full text available: pdf(982.37 KB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

Recently researchers have proposed modeling register allocation as an integer linear programming (IP) problem and solving it optimally for general purpose processors [17, 20] and for dedicated embedded systems [23]. Compared with traditional graph-coloring approaches, the IP-based allocators can improve a program's performance. However, the solution times are much slower.This paper presents an IP-based optimal register allocator which is much faster than previous work. We present several local a ...

12 Automating commutativity analysis at the design level

Greg Dennis, Robert Seater, Derek Rayside, Daniel Jackson
July 2004 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 2004 ACM SIGSOFT international symposium on Software testing and analysis**, Volume 29 Issue 4

Full text available: pdf(129.59 KB) Additional Information: full citation, abstract, references, index terms

Two operations commute if executing them serially in either order results in the same change of state. In a system in which commands may be issued simultaneously by different users, lack of commutativity can result in unpredictable behaviour, even if the commands are serialized, because one user's command may be preempted by another's, and thus executed in an unanticipated state. This paper describes an automated approach to analyzing commutativity. The operations are expressed as constraints in ...

Keywords: OCL, alloy, case study, commutativity, concurrency, critical systems, formal specification, lightweight formal methods, model checking, proton therapy, radiation therapy, testing

13 Reorganizing global schedules for register allocation
Gang Chen, Michael D. Smith
May 1999 **Proceedings of the 13th international conference on Supercomputing**

Full text available: pdf(1.15 MB) Additional Information: full citation, references, citings, index terms

Keywords: instruction-level parallelism, register allocation, superblock scheduling

14 Supporting dynamic data structures on distributed-memory machines
Anne Rogers, Martin C. Carlisle, John H. Reppy, Laurie J. Hendren
March 1995 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 17 Issue 2

Full text available: pdf(2.05 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Compiling for distributed-memory machines has been a very active research area in recent years. Much of this work has concentrated on programs that use arrays as their primary data structures. To date, little work has been done to address the problem of supporting programs that use pointer-based dynamic data structures. The techniques developed for supporting SPMD execution of array-based programs rely on the fact that arrays are statically defined and directly addressable. Recursive data s ...

Keywords: dynamic data structures

15 Space-efficient implementation of nested parallelism
Girija J. Narlikar, Guy E. Blelloch
June 1997 **ACM SIGPLAN Notices , Proceedings of the sixth ACM SIGPLAN symposium on Principles and practice of parallel programming**, Volume 32 Issue 7

Full text available: pdf(1.38 MB) Additional Information: full citation, abstract, references, citings, index terms

Many of today's high level parallel languages support dynamic, fine-grained parallelism. These languages allow the user to expose all the parallelism in the program, which is typically of a much higher degree than the number of processors. Hence an efficient scheduling algorithm is required to assign computations to processors at runtime. Besides having low overheads and good load balancing, it is important for the scheduling algorithm to minimize the space usage of the parallel program. This pa ...

Keywords: dynamic scheduling, language implementation, multithreading, nested parallelism, space efficiency

16 Partially preemptible hash joins
Hwee Hwa Pang, Michael J. Carey, Miron Livny
June 1993 **ACM SIGMOD Record , Proceedings of the 1993 ACM SIGMOD international conference on Management of data**, Volume 22 Issue 2

Full text available: pdf(1.42 MB) Additional Information: full citation, abstract, references, citings, index terms

With the advent of real-time and goal-oriented database systems, priority scheduling is likely to be an important feature in future database management systems. A consequence of priority scheduling is that a transaction may lose its buffers to higher-priority transactions, and may be given additional memory when transactions leave the system. Due to their heavy reliance on main memory, hash joins are especially vulnerable to fluctuations in memory availability. Previous studies have propose ...

17 Code scheduling and register allocation in large basic blocks
J. R. Goodman, W.-C. Hsu
June 1988 **Proceedings of the 2nd international conference on Supercomputing**

Full text available: pdf(1.20 MB) Additional Information: full citation, abstract, references, citings, index terms

We discuss the issues about the interdependency between code scheduling and register allocation. We present two methods as solutions: (1) an integrated code scheduling technique; and (2) a DAG-driven register allocator. The integrated code scheduling method combines two scheduling techniques—one to reduce pipeline delays and the other to minimize register usage—into a single phase. By keeping track of the number of available registers, the scheduler can choose the appropriate sc ...

18 Efficient and safe-for-space closure conversion

Zhong Shao, Andrew W. Appel

January 2000 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 22 Issue 1

Full text available: pdf(336.90 KB) Additional Information: full citation, abstract, references, citings, index terms

Modern compilers often implement function calls (or returns) in two steps: first, a "closure" environment is properly installed to provide access for free variables in the target program fragment; second, the control is transferred to the target by a "jump with arguments (for results)." Closure conversion—which decides where and how to represent closures at runtime—is a crucial step in the compilation of functional languages. This paper presents a new alg ...

Keywords: callee-save registers, closure conversion, closure representation, compiler optimization, flow analysis, heap-based compilation, space safety

19 Space and time efficient execution of parallel irregular computations

Cong Fu, Tao Yang

June 1997 **ACM SIGPLAN Notices , Proceedings of the sixth ACM SIGPLAN symposium on Principles and practice of parallel programming**, Volume 32 Issue 7

Full text available: pdf(1.35 MB) Additional Information: full citation, abstract, references, citings, index terms

Solving problems of large sizes is an important goal for parallel machines with multiple CPU and memory resources. In this paper, issues of efficient execution of overhead-sensitive parallel irregular computation under memory constraints are addressed. The irregular parallelism is modeled by task dependence graphs with mixed granularities. The trade-off in achieving both time and space efficiency is investigated. The main difficulty of designing efficient run-time system support is caused by the ...

20 Query evaluation techniques for large databases

Goetz Graefe

June 1993 **ACM Computing Surveys (CSUR)**, Volume 25 Issue 2

Full text available: pdf(9.37 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Database management systems will continue to manage large data volumes. Thus, efficient algorithms for accessing and manipulating large sets and sequences will be required to provide acceptable performance. The advent of object-oriented and extensible database systems will not solve this problem. On the contrary, modern data models exacerbate the problem: In order to manipulate large sets of complex objects as efficiently as today's database systems manipulate simple records, query-processi ...

Keywords: complex query evaluation plans, dynamic query evaluation plans, extensible database systems, iterators, object-oriented database systems, operator model of parallelization, parallel algorithms, relational database systems, set-matching algorithms, sort-hash duality

Terms used
journal and allocate or allocation or deallocate or deallocation and transaction priority

Found 21,006 of 145,831

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: 1 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 Provably efficient scheduling for languages with fine-grained parallelism

Guy E. Blelloch, Phillip B. Gibbons, Yossi Matias

March 1999 **Journal of the ACM (JACM)**, Volume 46 Issue 2

Full text available: pdf(321.43 KB) Additional Information: full citation, abstract, references, citings, index terms

Many high-level parallel programming languages allow for fine-grained parallelism. As in the popular work-time framework for parallel algorithm design, programs written in such languages can express the full parallelism in the program without specifying the mapping of program tasks to processors. A common concern in executing such programs is to schedule tasks to processors dynamically so as to minimize not only the execution time, but also the amount of space (memory) needed. Without caref ...

2 A new "quad-tree-based" sub-system allocation technique for mesh-connected parallel machines

Jeeraporn Srisawat, Nikitas A. Alexandridis

May 1999 **Proceedings of the 13th international conference on Supercomputing**

Full text available: pdf(1.22 MB) Additional Information: full citation, references, index terms

Keywords: mesh-connected parallel machines, parallel systems, quad-trees, resource allocation/deallocation, system partitioning

3 Processor allocation for a class of hypercube-like supercomputers

N. G. Haravu, S. G. Ziavras

December 1992 **Proceedings of the 1992 ACM/IEEE conference on Supercomputing**

Full text available: pdf(852.24 KB) Additional Information: full citation, references, citings, index terms

4 Integrating object-oriented programming and protected objects in Ada 95

A. J. Wellings, B. Johnson, B. Sanden, J. Kienzle, T. Wolf, S. Michell

May 2000 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 22 Issue 3

Full text available: pdf(245.47 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Integrating concurrent and object-oriented programming has been an active research topic since the late 1980's. There is a now a plethora of methods for achieving this integration. The majority of approaches have taken a sequential object-oriented language and made it concurrent. A few approaches have taken a concurrent language and made it object-oriented. The most important of this latter class is the Ada 95 language, which is an extension to the object-based concurrent programming langua ...

Keywords: Ada 95, concurrency, concurrent object-oriented programming, inheritance anomaly

5 Space-efficient scheduling of nested parallelism

Girija J. Narlikar, Guy E. Blelloch

January 1999 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 21 Issue 1

Full text available: Additional Information:

pdf(481.02 KB) full citation, abstract, references, citings, index terms, review

Many of today's high-level parallel languages support dynamic, fine-grained parallelism. These languages allow the user to expose all the parallelism in the program, which is typically of a much higher degree than the number of processors. Hence an efficient scheduling algorithm is required to assign computations to processors at runtime. Besides having low overheads and good load balancing, it is important for the scheduling algorithm to minimize the space usage of the parallel program. T ...

Keywords: dynamic scheduling, multithreading, nested parallelism, parallel language implementation, space efficiency

6 Integrating object-oriented programming and protected objects in Ada 95

A. J. Wellings, B. Johnson, B. Sanden, J. Kienzle, T. Wolf, S. Michell

June 2002 **ACM SIGAda Ada Letters**, Volume XXII Issue 2

Full text available: pdf(2.09 MB) Additional Information: full citation, abstract, references, citings, index terms

Integrating concurrent and object-oriented programming has been an active research topic since the late 1980's. There is now a plethora of methods for achieving this integration. The majority of approaches have taken a sequential object-oriented language and made it concurrent. A few approaches have taken a concurrent language and made it object-oriented. The most important of this latter class is the Ada 95 language, which is an extension to the object-based concurrent programming language Ada ...

Keywords: Ada 95, concurrency, concurrent object-oriented programming, inheritance anomaly

7 Space/time-efficient scheduling and execution of parallel irregular computations

Tao Yang, Cong Fu

November 1998 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 20 Issue 6

Full text available: pdf(374.95 KB) Additional Information: full citation, abstract, references, citings, index terms

In this article we investigate the trade-off between time and space efficiency in scheduling and executing parallel irregular computations on distributed-memory machines. We employ acyclic task dependence graphs to model irregular parallelism with mixed granularity, and we use direct remote memory access to support fast communication. We propose new scheduling techniques and a run-time active memory management scheme to improve memory utilization while retaining good time efficiency, and we ...

Keywords: DAG scheduling, direct remote memory access, irregular parallelism, run-time support

8 Efficient implementation of the first-fit strategy for dynamic storage allocation

R. P. Brent

July 1989 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 11 Issue 3

Full text available: pdf(1.05 MB) Additional Information: full citation, abstract, references, citings, index terms, review

We describe an algorithm that efficiently implements the first-fit strategy for dynamic storage allocation. The algorithm imposes a storage overhead of only one word per allocated block (plus a few percent of the total space used for dynamic storage), and the time required to allocate or free a block is O(log W), where W is the maximum number of words allocated dynamically. The algorithm is faster than many commonly used algorithms, especia ...

9 On local register allocation

Martin Farach, Vincenzo Liberatore

January 1998 **Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms**

Full text available: pdf(1.03 MB) Additional Information: full citation, references, citings, index terms

10 Space-efficient scheduling of parallelism with synchronization variables

Guy E. Blelloch, Phillip B. Gibbons, Girija J. Narlikar, Yossi Matias

June 1997 **Proceedings of the ninth annual ACM symposium on Parallel algorithms and architectures**

Full text available: pdf(1.67 MB) Additional Information: full citation, references, citings, index terms

11 Compilation and run-time systems: A faster optimal register allocator

Changqing Fu, Kent Wilken
November 2002 **Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture**

Full text available: pdf(582.37 KB) Additional Information: full citation, abstract, references, citings, index terms
Publisher Site

Recently researchers have proposed modeling register allocation as an integer linear programming (IP) problem and solving it optimally for general purpose processors [17, 20] and for dedicated embedded systems [23]. Compared with traditional graph-coloring approaches, the IP-based allocators can improve a program's performance. However, the solution times are much slower.This paper presents an IP-based optimal register allocator which is much faster than previous work. We present several local a ...

12 Automating commutativity analysis at the design level
Greg Dennis, Robert Seater, Derek Rayside, Daniel Jackson
July 2004 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 2004 ACM SIGSOFT international symposium on Software testing and analysis**, Volume 29 Issue 4
Full text available: pdf(126.59 KB) Additional Information: full citation, abstract, references, index terms

Two operations commute if executing them serially in either order results in the same change of state. In a system in which commands may be issued simultaneously by different users, lack of commutativity can result in unpredictable behaviour, even if the commands are serialized, because one user's command may be preempted by another's, and thus executed in an unanticipated state. This paper describes an automated approach to analyzing commutativity. The operations are expressed as constraints in ...

Keywords: OCL, alloy, case study, commutativity, concurrency, critical systems, formal specification, lightweight formal methods, model checking, proton therapy, radiation therapy, testing

13 Reorganizing global schedules for register allocation
Gang Chen, Michael D. Smith
May 1999 **Proceedings of the 13th international conference on Supercomputing**
Full text available: pdf(1.15 MB) Additional Information: full citation, references, citings, index terms

Keywords: instruction-level parallelism, register allocation, superblock scheduling

14 Supporting dynamic data structures on distributed-memory machines
Anne Rogers, Martin C. Carlisle, John H. Reppy, Laurie J. Hendren
March 1995 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 17 Issue 2
Full text available: pdf(2.05 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Compiling for distributed-memory machines has been a very active research area in recent years. Much of this work has concentrated on programs that use arrays as their primary data structures. To date, little work has been done to address the problem of supporting programs that use pointer-based dynamic data structures. The techniques developed for supporting SPMD execution of array-based programs rely on the fact that arrays are statically defined and directly addressable. Recursive data s ...

Keywords: dynamic data structures

15 Space-efficient implementation of nested parallelism
Girija J. Narlikar, Guy E. Blelloch
June 1997 **ACM SIGPLAN Notices , Proceedings of the sixth ACM SIGPLAN symposium on Principles and practice of parallel programming**, Volume 32 Issue 7
Full text available: pdf(1.38 MB) Additional Information: full citation, abstract, references, citings, index terms

Many of today's high level parallel languages support dynamic, fine-grained parallelism. These languages allow the user to expose all the parallelism in the program, which is typically of a much higher degree than the number of processors. Hence an efficient scheduling algorithm is required to assign computations to processors at runtime. Besides having low overheads and good load balancing, it is important for the scheduling algorithm to minimize the space usage of the parallel program. This pa ...

Keywords: dynamic scheduling, language implementation, multithreading, nested parallelism, space efficiency

16 Partially preemptible hash joins

Hwee Hwa Pang, Michael J. Carey, Miron Livny

June 1993 **ACM SIGMOD Record , Proceedings of the 1993 ACM SIGMOD international conference on Management of data**, Volume 22 Issue 2

Full text available: pdf(1.42 MB) Additional Information: full citation, abstract, references, citings, index terms

With the advent of real-time and goal-oriented database systems, priority scheduling is likely to be an important feature in future database management systems. A consequence of priority scheduling is that a transaction may lose its buffers to higher-priority transactions, and may be given additional memory when transactions leave the system. Due to their heavy reliance on main memory, hash joins are especially vulnerable to fluctuations in memory availability. Previous studies have propose ...

17 Code scheduling and register allocation in large basic blocks

J. R. Goodman, W.-C. Hsu

June 1988 **Proceedings of the 2nd international conference on Supercomputing**

Full text available: pdf(1.20 MB) Additional Information: full citation, abstract, references, citings, index terms

We discuss the issues about the interdependency between code scheduling and register allocation. We present two methods as solutions: (1) an integrated code scheduling technique; and (2) a DAG-driven register allocator. The integrated code scheduling method combines two scheduling techniques—one to reduce pipeline delays and the other to minimize register usage—into a single phase. By keeping track of the number of available registers, the scheduler can choose the appropriate sc ...

18 Efficient and safe-for-space closure conversion

Zhong Shao, Andrew W. Appel

January 2000 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 22 Issue 1

Full text available: pdf(336.80 KB) Additional Information: full citation, abstract, references, citings, index terms

Modern compilers often implement function calls (or returns) in two steps: first, a "closure" environment is properly installed to provide access for free variables in the target program fragment; second, the control is transferred to the target by a "jump with arguments (for results)." Closure conversion—which decides where and how to represent closures at runtime—is a crucial step in the compilation of functional languages. This paper presents a new alg ...

Keywords: callee-save registers, closure conversion, closure representation, compiler optimization, flow analysis, heap-based compilation, space safety

19 Space and time efficient execution of parallel irregular computations

Cong Fu, Tao Yang

June 1997 **ACM SIGPLAN Notices , Proceedings of the sixth ACM SIGPLAN symposium on Principles and practice of parallel programming**, Volume 32 Issue 7

Full text available: pdf(1.35 MB) Additional Information: full citation, abstract, references, citings, index terms

Solving problems of large sizes is an important goal for parallel machines with multiple CPU and memory resources. In this paper, issues of efficient execution of overhead-sensitive parallel irregular computation under memory constraints are addressed. The irregular parallelism is modeled by task dependence graphs with mixed granularities. The trade-off in achieving both time and space efficiency is investigated. The main difficulty of designing efficient run-time system support is caused by the ...

20 Query evaluation techniques for large databases

Goetz Graefe

June 1993 **ACM Computing Surveys (CSUR)**, Volume 25 Issue 2

Full text available: pdf(9.37 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Database management systems will continue to manage large data volumes. Thus, efficient algorithms for accessing and manipulating large sets and sequences will be required to provide acceptable performance. The advent of object-oriented and extensible database systems will not solve this problem. On the contrary, modern data models exacerbate the problem: In order to manipulate large sets of complex objects as efficiently as today's database systems manipulate simple records, query-processi ...

Keywords: complex query evaluation plans, dynamic query evaluation plans, extensible database systems, iterators, object-oriented database systems, operator model of parallelization, parallel algorithms, relational database systems, set-matching algorithms, sort-hash duality

Terms used **journal space** and **allocate** or **allocation** or **deallocate** or **deallocation** and **transaction priority**

Found **15,675** of **145,831**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 A new "quad-tree-based" sub-system allocation technique for mesh-connected parallel machines

Jeeraporn Srisawat, Nikitas A. Alexandridis

May 1999 **Proceedings of the 13th international conference on Supercomputing**

Full text available: pdf(1.22 MB) Additional Information: full citation, references, index terms

Keywords: mesh-connected parallel machines, parallel systems, quad-trees, resource allocation/deallocation, system partitioning

2 Provably efficient scheduling for languages with fine-grained parallelism

Guy E. Blelloch, Phillip B. Gibbons, Yossi Matias

March 1999 **Journal of the ACM (JACM)**, Volume 46 Issue 2

Full text available: pdf(321.43 KB) Additional Information: full citation, abstract, references, citings, index terms

Many high-level parallel programming languages allow for fine-grained parallelism. As in the popular work-time framework for parallel algorithm design, programs written in such languages can express the full parallelism in the program without specifying the mapping of program tasks to processors. A common concern in executing such programs is to schedule tasks to processors dynamically so as to minimize not only the execution time, but also the amount of space (memory) needed. Without caref ...

3 Processor allocation for a class of hypercube-like supercomputers

N. G. Haravu, S. G. Ziavras

December 1992 **Proceedings of the 1992 ACM/IEEE conference on Supercomputing**

Full text available: pdf(852.24 KB) Additional Information: full citation, references, citings, index terms

4 Integrating object-oriented programming and protected objects in Ada 95

A. J. Wellings, B. Johnson, B. Sanden, J. Kienzle, T. Wolf, S. Michell

May 2000 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 22 Issue 3

Full text available: pdf(245.47 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Integrating concurrent and object-oriented programming has been an active research topic since the late 1980's. There is a now a plethora of methods for achieving this integration. The majority of approaches have taken a sequential object-oriented language and made it concurrent. A few approaches have taken a concurrent language and made it object-oriented. The most important of this latter class is the Ada 95 language, which is an extension to the object-based concurrent programming langua ...

Keywords: Ada 95, concurrency, concurrent object-oriented programming, inheritance anomaly

5 Space-efficient scheduling of nested parallelism

Girija J. Narlikar, Guy E. Blelloch

January 1999 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 21 Issue 1

Full text available: pdf(481.02 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Many of today's high-level parallel languages support dynamic, fine-grained parallelism. These languages allow the user to expose all the parallelism in the program, which is typically of a much higher degree than the number of processors. Hence an efficient scheduling algorithm is required to assign computations to processors at runtime. Besides having low overheads and good load balancing, it is important for the scheduling algorithm to minimize the space usage of the parallel program. T ...

Keywords: dynamic scheduling, multithreading, nested parallelism, parallel language implementation, space efficiency

6 Call-cost directed register allocation

Guei-Yuan Lueh, Thomas Gross

May 1997 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1997 conference on Programming language design and implementation**, Volume 32 Issue 5

Full text available: pdf(2.01 MB) Additional Information: full citation, abstract, references, citings, index terms

Choosing the right kind of register for a live range plays a major role in eliminating the register-allocation overhead when the compiled function is frequently executed or function tails are on the most frequently executed paths. Picking the wrong kind of register for a live range incurs a high penalty that may dominate the total overhead of register allocation. In this paper, we present three improvements, storage-class analysis, benefit-driven simplification, and preference decision that are ...

7 Integrating object-oriented programming and protected objects in Ada 95

A. J. Wellings, B. Johnson, B. Sanden, J. Kienzle, T. Wolf, S. Michell

June 2002 **ACM SIGAda Ada Letters**, Volume XXII Issue 2

Full text available: pdf(2.09 MB) Additional Information: full citation, abstract, references, citings, index terms

Integrating concurrent and object-oriented programming has been an active research topic since the late 1980's. There is now a plethora of methods for achieving this integration. The majority of approaches have taken a sequential object-oriented language and made it concurrent. A few approaches have taken a concurrent language and made it object-oriented. The most important of this latter class is the Ada 95 language, which is an extension to the object-based concurrent programming language Ada ...

Keywords: Ada 95, concurrency, concurrent object-oriented programming, inheritance anomaly

8 Efficient implementation of the first-fit strategy for dynamic storage allocation

R. P. Brent

July 1989 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 11 Issue 3

Full text available: pdf(1.05 MB) Additional Information: full citation, abstract, references, citings, index terms, review

We describe an algorithm that efficiently implements the first-fit strategy for dynamic storage allocation. The algorithm imposes a storage overhead of only one word per allocated block (plus a few percent of the total space used for dynamic storage), and the time required to allocate or free a block is O(log W), where W is the maximum number of words allocated dynamically. The algorithm is faster than many commonly used algorithms, especia ...

9 Comparing the reliability provided by tasks or protected objects for implementing a resource allocation service: a case study

C. Kaiser, J. F. Pradat-Peyre

November 1997 **Proceedings of the conference on TRI-Ada '97**

Full text available: pdf(1.53 MB) Additional Information: full citation, references, citings, index terms

10 Compilation and run-time systems: A faster optimal register allocator

Changqing Fu, Kent Wilken

November 2002 **Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture**

Full text available: pdf(982.37 KB) Additional Information: full citation, abstract, references, citings, index terms

Publisher Site

Recently researchers have proposed modeling register allocation as an integer linear programming (IP) problem and solving it optimally for general purpose processors [17, 20] and for dedicated embedded systems [23]. Compared with traditional graph-coloring approaches, the IP-based allocators can improve a program's performance. However, the solution times are much slower.This

paper presents an IP-based optimal register allocator which is much faster than previous work. We present several local a ...

[11] Scalable lock-free dynamic memory allocation

Maged M. Michael

June 2004 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2004 conference on Programming language design and implementation**, Volume 39 Issue 6

Full text available: pdf(213.34 KB) Additional Information: full citation, abstract, references, index terms

Dynamic memory allocators (malloc/free) rely on mutual exclusion locks for protecting the consistency of their shared data structures under multithreading. The use of locking has many disadvantages with respect to performance, availability, robustness, and programming flexibility. A lock-free memory allocator guarantees progress regardless of whether some threads are delayed or even killed and regardless of scheduling policies. This paper presents a completely lock-free memory allocator. It uses ...

Keywords: async-signal-safe, availability, lock-free, malloc

[12] Automating commutativity analysis at the design level

Greg Dennis, Robert Seater, Derek Rayside, Daniel Jackson

July 2004 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 2004 ACM SIGSOFT international symposium on Software testing and analysis**, Volume 29 Issue 4

Full text available: pdf(129.59 KB) Additional Information: full citation, abstract, references, index terms

Two operations commute if executing them serially in either order results in the same change of state. In a system in which commands may be issued simultaneously by different users, lack of commutativity can result in unpredictable behaviour, even if the commands are serialized, because one user's command may be preempted by another's, and thus executed in an unanticipated state. This paper describes an automated approach to analyzing commutativity. The operations are expressed as constraints in ...

Keywords: OCL, alloy, case study, commutativity, concurrency, critical systems, formal specification, lightweight formal methods, model checking, proton therapy, radiation therapy, testing

[13] Design of a Machine-Independent Optimizing System for Emulator Development

Perng-Ti Ma, T. G. Lewis

April 1980 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 2 Issue 2

Full text available: pdf(1.32 MB) Additional Information: full citation, abstract, references, citings, index terms

Methods are described to translate a certain machine-independent intermediate language (IML) to efficient microprograms for a class of horizontal microprogrammable machines. The IML is compiled directly from a high-level microprogramming language used to implement a virtual instruction set processor as a microprogram. The primary objective of the IML-to-host machine interface design is to facilitate language portability. Transportability is accomplished by use of a field descript ...

[14] A structural view of the Cedar programming environment

Daniel C. Swinehart, Polle T. Zellweger, Richard J. Beach, Robert B. Hagmann

August 1986 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 8 Issue 4

Full text available: pdf(6.32 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents an overview of the Cedar programming environment, focusing on its overall structure—that is, the major components of Cedar and the way they are organized. Cedar supports the development of programs written in a single programming language, also called Cedar. Its primary purpose is to increase the productivity of programmers whose activities include experimental programming and the development of prototype software systems for a high-performance personal computer. T ...

[15] Supporting dynamic data structures on distributed-memory machines

Anne Rogers, Martin C. Carlisle, John H. Reppy, Laurie J. Hendren

March 1995 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 17 Issue 2

Full text available: pdf(2.05 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Compiling for distributed-memory machines has been a very active research area in recent years. Much of this work has concentrated on programs that use arrays as their primary data structures. To date, little work has been done to address the problem of supporting programs that use pointer-based dynamic data structures. The techniques developed for supporting SPMD execution of array-based programs rely on the fact that arrays are statically defined and directly addressable. Recursive data s ...

Keywords: dynamic data structures

[16] Space/time-efficient scheduling and execution of parallel irregular computations
Tao Yang, Cong Fu
November 1998 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 20 Issue 6
Full text available: pdf(374.95 KB) Additional Information: full citation, abstract, references, citings, index terms

In this article we investigate the trade-off between time and space efficiency in scheduling and executing parallel irregular computations on distributed-memory machines. We employ acyclic task dependence graphs to model irregular parallelism with mixed granularity, and we use direct remote memory access to support fast communication. We propose new scheduling techniques and a run-time active memory management scheme to improve memory utilization while retaining good time efficiency, and we ...

Keywords: DAG scheduling, direct remote memory access, irregular parallelism, run-time support

[17] Reorganizing global schedules for register allocation
Gang Chen, Michael D. Smith
May 1999 **Proceedings of the 13th international conference on Supercomputing**
Full text available: pdf(1.15 MB) Additional Information: full citation, references, citings, index terms

Keywords: instruction-level parallelism, register allocation, superblock scheduling

[18] Real time resource allocation in distributed systems
John Reif, Paul Spirakis
August 1982 **Proceedings of the first ACM SIGACT-SIGOPS symposium on Principles of distributed computing**
Full text available: pdf(852.77 KB) Additional Information: full citation, abstract, references, citings, index terms

In this paper we consider a resource allocation problem which is local in the sense that the maximum number of users competing for a particular resource at any time instant is bounded and also at any time instant the maximum number of resources that a user is willing to get is bounded. The problem may be viewed as that of achieving matchings in dynamically changing hypergraphs, via a distributed algorithm. We show that this problem is related to the fundamental problem of < ...

[19] Fbufs: a high-bandwidth cross-domain transfer facility
Peter Druschel, Larry L. Peterson
December 1993 **ACM SIGOPS Operating Systems Review , Proceedings of the fourteenth ACM symposium on Operating systems principles**, Volume 27 Issue 5
Full text available: pdf(1.35 MB) Additional Information: full citation, abstract, references, citings, index terms

We have designed and implemented a new operating system facility for I/O buffer management and data transferacross protection domain boundaries on shared memory machines. This facility, called *fast buffers* (fbufs), combines virtual page remapping with shared virtual memory, and exploits locality in I/O traffic to achieve high throughput without compromising protection, security, or modularity. goal is to help deliver the high bandwidth afforded by emerging high-speed networks to user-leve ...

[20] On local register allocation
Martin Farach, Vincenzo Liberatore
January 1998 **Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms**
Full text available: pdf(1.03 MB) Additional Information: full citation, references, citings, index terms

Terms used **journal space** and **memory allocation** or **memory deallocation** and **transaction priority**

Found **13,702** of **145,831**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 Provably efficient scheduling for languages with fine-grained parallelism

Guy E. Blelloch, Phillip B. Gibbons, Yossi Matias

March 1999 **Journal of the ACM (JACM)**, Volume 46 Issue 2

Full text available: pdf(321.43 KB) Additional Information: full citation, abstract, references, citings, index terms

Many high-level parallel programming languages allow for fine-grained parallelism. As in the popular work-time framework for parallel algorithm design, programs written in such languages can express the full parallelism in the program without specifying the mapping of program tasks to processors. A common concern in executing such programs is to schedule tasks to processors dynamically so as to minimize not only the execution time, but also the amount of space (memory) needed. Without caref ...

2 Space-efficient scheduling of parallelism with synchronization variables

Guy E. Blelloch, Phillip B. Gibbons, Girija J. Narlikar, Yossi Matias

June 1997 **Proceedings of the ninth annual ACM symposium on Parallel algorithms and architectures**

Full text available: pdf(1.67 MB) Additional Information: full citation, references, citings, index terms

3 A performance study of the cancelback protocol for Time Warp

Samir R. Das, Richard M. Fujimoto

July 1993 **ACM SIGSIM Simulation Digest , Proceedings of the seventh workshop on Parallel and distributed simulation**, Volume 23 Issue 1

Full text available: pdf(1.09 MB) Additional Information: full citation, abstract, references, citings, index terms

This work presents results from an experimental evaluation of the space-time tradeoffs in Time Warp augmented with the cancelback protocol for memory management. An implementation of the cancelback protocol on Time Warp is described that executes on a shared memory multiprocessor, a 32 processor Kendall Square Research Machine (KSR1). The implementation supports canceling back more than one object when memory has been exhausted. The limited memory performance of the system is evaluated for ...

4 The priority-based coloring approach to register allocation

Fred C. Chow, John L. Hennessy

October 1990 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 12 Issue 4

Full text available: pdf(2.97 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Global register allocation plays a major role in determining the efficacy of an optimizing compiler. Graph coloring has been used as the central paradigm for register allocation in modern compilers. A straightforward coloring approach can suffer from several shortcomings. These shortcomings are addressed in this paper by coloring the graph using a priority ordering. A natural method for dealing with the spilling emerges from this approach. The detailed algorithms for a priority-based colori ...

5 A dynamic processor allocation policy for multiprogrammed shared-memory multiprocessors

Cathy McCann, Raj Vaswani, John Zahorjan

May 1993 **ACM Transactions on Computer Systems (TOCS)**, Volume 11 Issue 2

Full text available: pdf(2.26 MB) Additional Information: full citation, abstract, references, citings, index terms

We propose and evaluate empirically the performance of a dynamic processor-scheduling policy for multiprogrammed shared-memory multiprocessors. The policy is dynamic in that it reallocates

processors from one parallel job to another based on the currently realized parallelism of those jobs. The policy is suitable for implementation in production systems in that: —It interacts well with very efficient user-level thread packages, leaving to them many low-level thr ...

Keywords: shared memory parallel processors, threads, two-level scheduling

6 A resource management framework for priority-based physical-memory allocation

Kingsley Cheung, Gernot Heiser

January 2002 **Australian Computer Science Communications , Proceedings of the seventh Asia-Pacific conference on Computer systems architecture - Volume 6**, Volume 24 Issue 3

Full text available: pdf(1.32 MB) Additional Information: full citation, abstract, references, index terms

Most multitasking operating systems support scheduling priorities in order to ensure that processor time is allocated to important or time-critical processes in preference to less important ones. Ideally this would prevent a low-priority process from slowing the execution of a high-priority one. In practice, strict prioritisation is undermined by a lack of suitable allocation policy for resources other than CPU time. For example, a low priority process may degrade the execution speed of a high-p ...

7 Simultaneous reference allocation in code generation for dual data memory bank ASIPs

Ashok Sudarsanam, Sharad Malik

April 2000 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 5 Issue 2

Full text available: pdf(156.30 KB) Additional Information: full citation, abstract, references, citings, index terms

We address the problem of code generation for DSP systems on a chip. In such systems, the amount of silicon devoted of program ROM is limited, so application software must be sufficiently dense. Additionally, the software must be written so as to meet various high-performance constraints, which may include hard real-time constraints. Unfortunately, current compiler technology is unable to generate high-quality code for DSPs, whose architectures are highly irregular. Thus, designers often r ...

Keywords: code generation, code optimization, graph labelling, memory bank assignment, register allocation

8 External memory algorithms and data structures: dealing with massive data

Jeffrey Scott Vitter

June 2001 **ACM Computing Surveys (CSUR)**, Volume 33 Issue 2

Full text available: pdf(828.46 KB) Additional Information: full citation, abstract, references, citings, index terms

Data sets in large applications are often too massive to fit completely inside the computers internal memory. The resulting input/output communication (or I/O) between fast internal memory and slower external memory (such as disks) can be a major performance bottleneck. In this article we survey the state of the art in the design and analysis of external memory (or EM) algorithms and data structures, where the goal is to exploit locality in order to reduce the I/O costs. We consider a varie ...

Keywords: B-tree, I/O, batched, block, disk, dynamic, extendible hashing, external memory, hierarchical memory, multidimensional access methods, multilevel memory, online, out-of-core, secondary storage, sorting

9 Software transactional memory

Nir Shavit, Dan Touitou

August 1995 **Proceedings of the fourteenth annual ACM symposium on Principles of distributed computing**

Full text available: pdf(913.62 KB) Additional Information: full citation, references, citings, index terms

10 Static scheduling algorithms for allocating directed task graphs to multiprocessors

Yu-Kwong Kwok, Ishfaq Ahmad

December 1999 **ACM Computing Surveys (CSUR)**, Volume 31 Issue 4

Full text available: pdf(723.58 KB) Additional Information: full citation, abstract, references, citings, index terms

Static scheduling of a program represented by a directed task graph on a multiprocessor system to

minimize the program completion time is a well-known problem in parallel processing. Since finding an optimal schedule is an NP-complete problem in general, researchers have resorted to devising efficient heuristics. A plethora of heuristics have been proposed based on a wide spectrum of techniques, including branch-and-bound, integer-programming, searching, graph-theory, randomization, genetic ...

Keywords: DAG, automatic parallelization, multiprocessors, parallel processing, software tools, static scheduling, task graphs

[11] Fusion-based register allocation

Guei-Yuan Lueh, Thomas Gross, Ali-Reza Adl-Tabatabai

May 2000 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 22 Issue 3

Full text available: pdf(475.45 KB) Additional Information: full citation, abstract, references, citings, index terms, review

The register allocation phase of a compiler maps live ranges of a program to registers. If there are more candidates than there are physical registers, the register allocator must spill a live range (the home location is in memory) or split a live range (the live range occupies multiple locations). One of the challenges for a register allocator is to deal with spilling and splitting together. Fusion-based register allocation uses the structure of the program to make splitting and spilling d ...

Keywords: performance evaluation, register allocation

[12] A simple interprocedural register allocation algorithm and its effectiveness for LISP

Peter A. Steenkiste, John L. Hennessy

January 1989 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 11 Issue 1

Full text available: pdf(2.56 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Register allocation is an important optimization in many compilers, but with per-procedure register allocation, it is often not possible to make good use of a large register set. Procedure calls limit the improvement from global register allocation, since they force variables allocated to registers to be saved and restored. This limitation is more pronounced in LISP programs due to the higher frequency of procedure calls. An interprocedural register allocation algorithm is developed by simp ...

[13] Evaluating models of memory allocation

Benjamin Zorn, Dirk Grunwald

January 1994 **ACM Transactions on Modeling and Computer Simulation (TOMACS)**, Volume 4 Issue 1

Full text available: pdf(1.69 MB) Additional Information: full citation, abstract, references, citings, index terms

Because dynamic memory management is an important part of a large class of computer programs, high-performance algorithms for dynamic memory management have been and will continue to be of considerable interest. The goal of this research is to explore the size and accuracy of synthetic models of program allocation behavior. These models, if accurate enough, proved an attractive alternative to algorithm evaluation based on trace-driven simulation using actual traces. Based on our analysis, w ...

Keywords: dynamic storage allocation, model evaluation, program behavior modeling, program measurement, trace-driven simulation

[14] Managing memory for real-time queries

Hwee Hwa Pang, Michael J. Carey, Miron Livny

May 1994 **ACM SIGMOD Record , Proceedings of the 1994 ACM SIGMOD international conference on Management of data**, Volume 23 Issue 2

Full text available: pdf(1.59 MB) Additional Information: full citation, abstract, references, citings, index terms

The demanding performance objectives that real-time database systems (RTDBS) face necessitate the use of priority resource scheduling. This paper introduces a Priority Memory Management (PMM) algorithm that is designed to schedule queries in RTDBS. PMM attempts to minimize the number of missed deadlines by adapting both its multiprogramming level and its memory allocation strategy to the characteristics of the offered workload. A series of simulation experiments confirms th ...

[15] Applying priorities to memory allocation

Sven G. Robertz

June 2002 **ACM SIGPLAN Notices , Proceedings of the 3rd international symposium on Memory management**, Volume 38 Issue 2 supplement

Full text available: pdf(618.64 KB) Additional Information: full citation, abstract, references, index terms

A novel approach of applying priorities to memory allocation is presented and it is shown how this can be used to enhance the robustness of real-time applications. The proposed mechanisms can also be

used to increase performance of systems with automatic memory management by limiting the amount of garbage collection work.A way of introducing priorities for memory allocation in a Java system without making any changes to the syntax of the language is proposed and this has been implemented in an e ...

Keywords: embedded systems, memory allocation, real-time garbage collection, robustness

16 Scalable lock-free dynamic memory allocation

Maged M. Michael

June 2004 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2004 conference on Programming language design and implementation**, Volume 39 Issue 6

Full text available: pdf(213.94 KB) Additional Information: full citation, abstract, references, index terms

Dynamic memory allocators (malloc/free) rely on mutual exclusion locks for protecting the consistency of their shared data structures under multithreading. The use of locking has many disadvantages with respect to performance, availability, robustness, and programming flexibility. A lock-free memory allocator guarantees progress regardless of whether some threads are delayed or even killed and regardless of scheduling policies. This paper presents a completely lock-free memory allocator. It uses ...

Keywords: async-signal-safe, availability, lock-free, malloc

17 Improvements to graph coloring register allocation

Preston Briggs, Keith D. Cooper, Linda Torczon

May 1994 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 16 Issue 3

Full text available: pdf(2.00 MB) Additional Information: full citation, abstract, references, citings, index terms, review

We describe two improvements to Chaitin-style graph coloring register allocators. The first, optimistic coloring, uses a stronger heuristic to find a k-coloring for the interference graph. The second extends Chaitin's treatment of rematerialization to handle a larger class of values. These techniques are complementary. Optimistic coloring decreases the number of procedures that require spill code and reduces the amount of spill code when sp ...

Keywords: code generation, graph coloring, register allocation

18 Efficient register allocation via coloring using clique separators

Rajiv Gupta, Mary Lou Soffa, Denise Ombres

May 1994 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 16 Issue 3

Full text available: pdf(1.16 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Although graph coloring is widely recognized as an effective technique for register allocation, memory demands can become quite high for large interference graphs that are needed in coloring. In this paper we present an algorithm that uses the notion of clique separators to improve the space overhead of coloring. The algorithm, based on a result by R. Tarjan regarding the colorability of graphs, partitions program code into code segments using clique separators. The interference graphs for ...

Keywords: clique separators, graph coloring, interference graph, node priorities, spans, spill code

19 On-chip communication architectures: analysis and optimisation: Efficient exploration of on-chip bus architectures and memory allocation

Sungchan Kim, Chaeseok Im, Soonhoi Ha

September 2004 **Proceedings of the 2nd IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis**

Full text available: pdf(594.85 KB) Additional Information: full citation, abstract, references, index terms

Separation between computation and communication in system design allows the system designer to explore the communication architecture independently of component selection and mapping. In this paper we present an iterative two-step exploration methodology for bus-based on-chip communication architecture and memory allocation, assuming that memory traces from the processing elements are given from the mapping stage. The proposed method uses a static performance estimation technique to reduce the ...

Keywords: communication architecture optimization, design space exploration, memory allocation, system-on-a-chip

20 A parallel execution model for a database machine with high performances

Didier Donsez, Pascal Faudemay

July 1990 **Proceedings of the second international symposium on Databases in parallel and distributed systems**

Full text available: pdf(1.47 MB) Additional Information: full citation, abstract, references, index terms

In this paper, we present a mixed MIMD / SIMD execution model for a reconfigurable computer. This model is adapted to the use of a specialized associative coprocessor, embedded in this host machine. A main characteristic of the model is that it uses four types of processes (decoding, calculus, coprocessor communication and transaction manager), and that in principle one process of each type is allowed on each processor. Time intervals are allocated to operations into partitions of t ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next